SUPERIOR UNIVERSITY

Pre-Mid(Task#2)

Submitted to: Sir Aqib

Submitted by: Hifza Khalid

Roll#:SU92_BSSEM_F22_202

Subject: Advance Computer Programming

Section: BSSE-4D

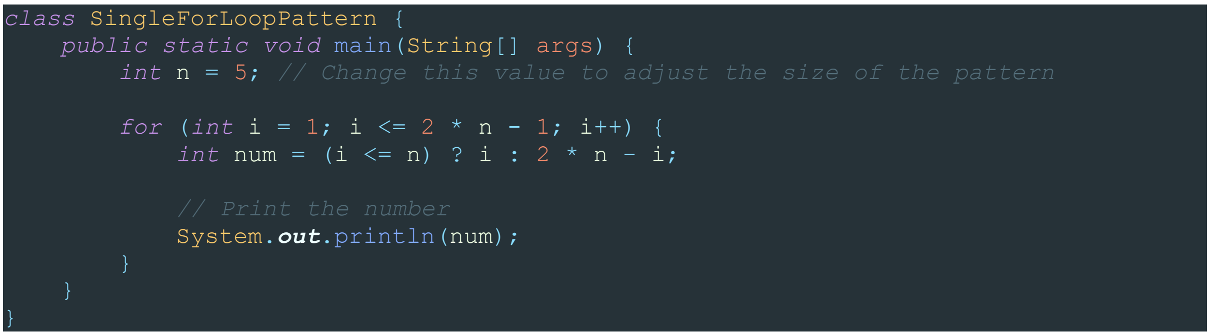
Date: Jan15,2024.

Topic: For Loop

Question#1:

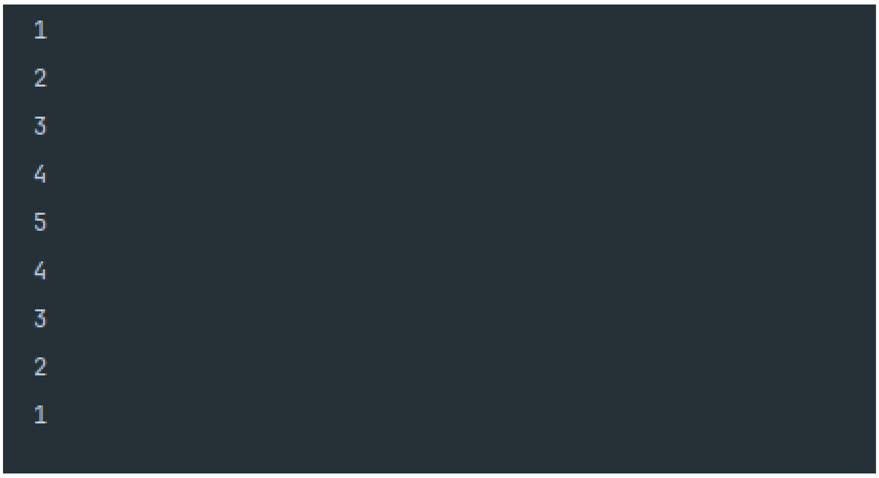
By For Loop print this:

1 2 3 4 5 4 3 2 1 _Use println for print each num in next line



```java
class SingleForLoopPattern {
    public static void main(String[] args) {
        int n = 5; // Change this value to adjust the size of the pattern

        for (int i = 1; i <= 2 * n - 1; i++) {
            int num = (i <= n) ? i : 2 * n - i;

            // Print the number
            System.out.println(num);
        }
    }
}
```

Output:

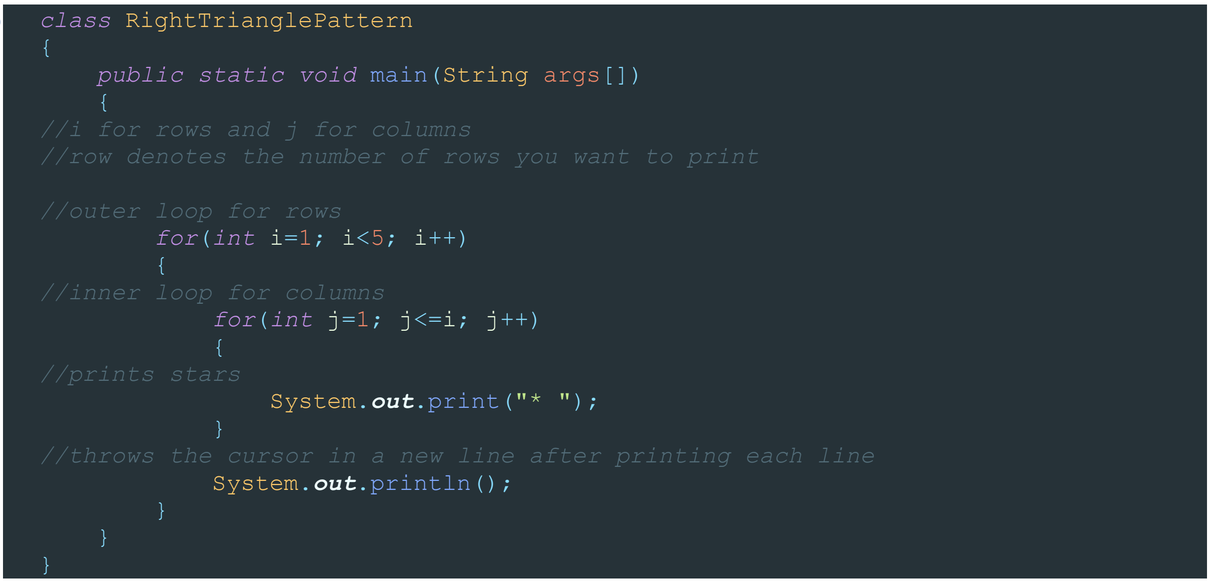


```
1
2
3
4
5
4
3
2
1
```

Question#2:

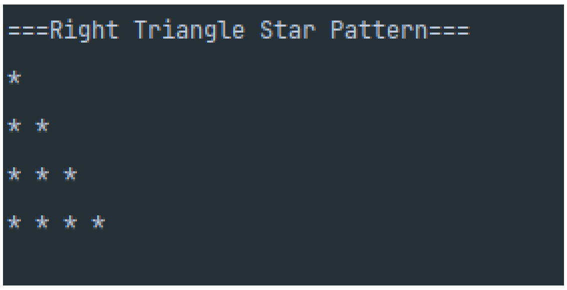
- **Java Pattern Print**
- **Start Pattern**

1.Right Triangle Star Pattern



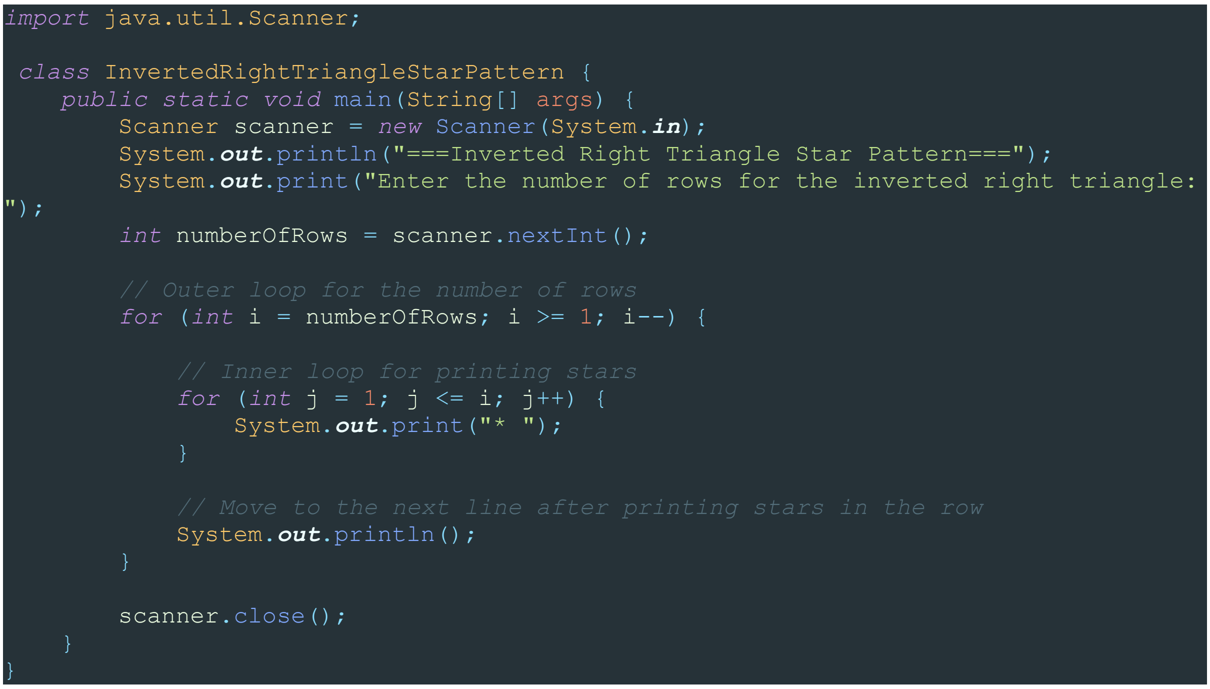
```
class RightTrianglePattern
{
    public static void main(String args[])
    {
//i for rows and j for columns
//row denotes the number of rows you want to print

//outer loop for rows
        for(int i=1; i<5; i++)
        {
//inner loop for columns
            for(int j=1; j<=i; j++)
            {
//prints stars
                System.out.print("* ");
            }
//throws the cursor in a new line after printing each line
            System.out.println();
        }
    }
}
```

Output:



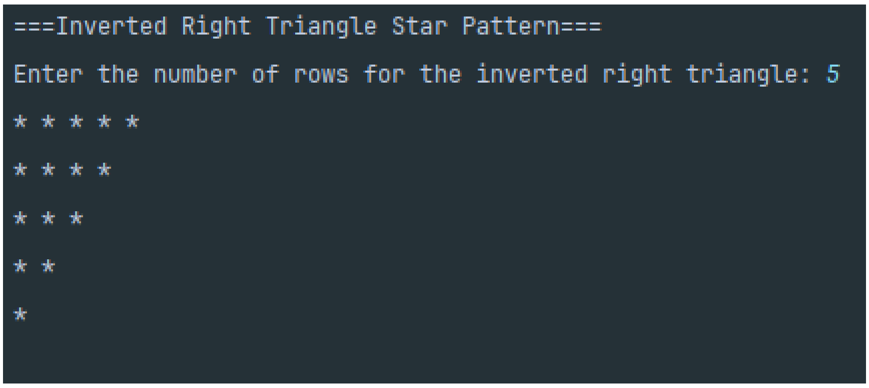
```
===Right Triangle Star Pattern===
*
* *
* * *
* * * *
```

2.Inverted Right Triangle Star Pattern



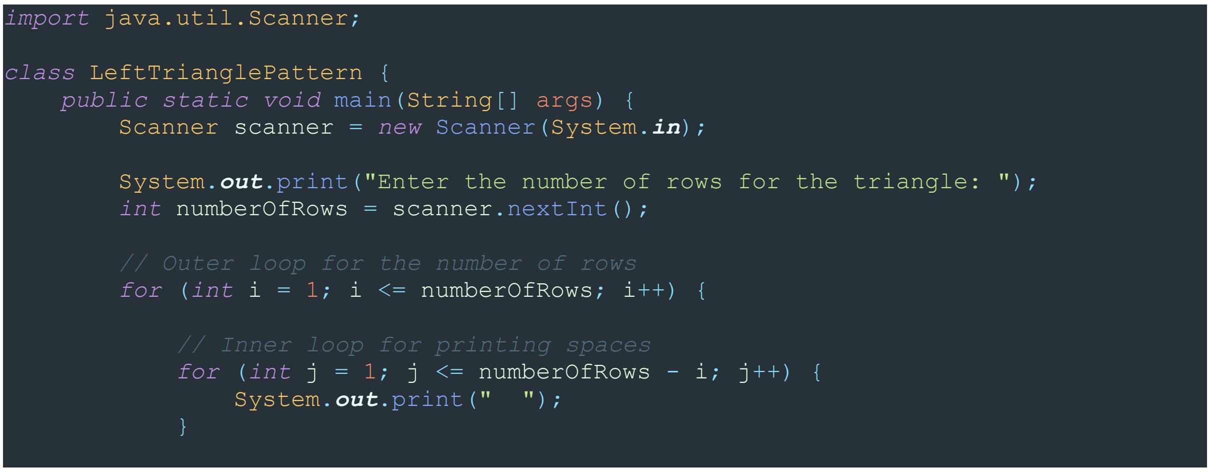
```java
import java.util.Scanner;

 class InvertedRightTriangleStarPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("===Inverted Right Triangle Star Pattern===");
        System.out.print("Enter the number of rows for the inverted right triangle:
");
        int numberOfRows = scanner.nextInt();

        // Outer loop for the number of rows
        for (int i = numberOfRows; i >= 1; i--) {

            // Inner loop for printing stars
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }

            // Move to the next line after printing stars in the row
            System.out.println();
        }

        scanner.close();
    }
}
```

Output:

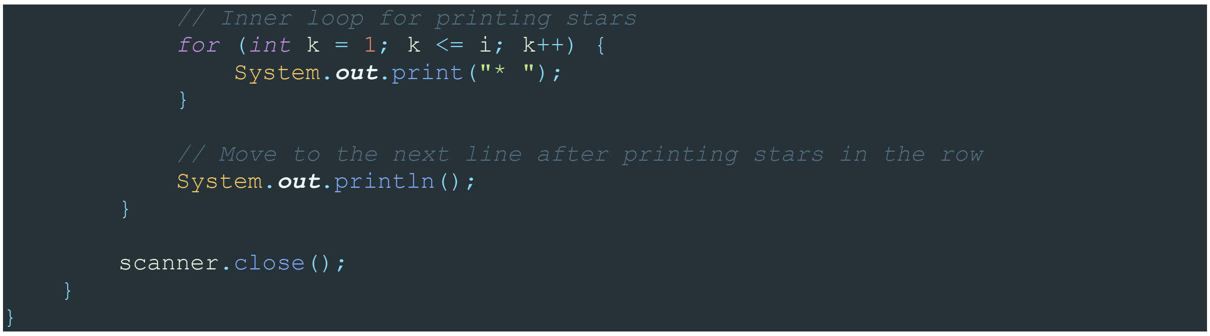


```
===Inverted Right Triangle Star Pattern===

Enter the number of rows for the inverted right triangle: 5

* * * * *

* * * *

* * *

* *

*
```

3. Left Triangle Star Pattern

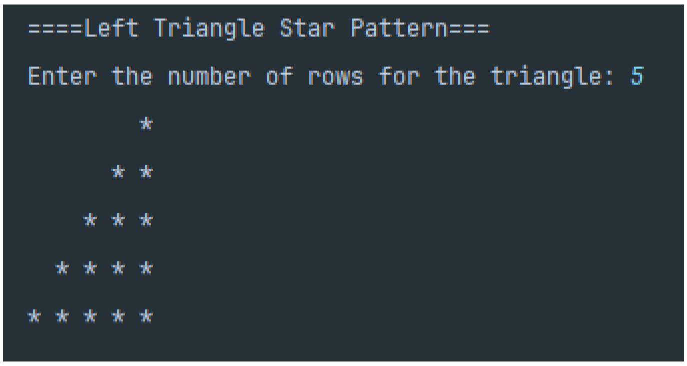


```java
import java.util.Scanner;

class LeftTrianglePattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of rows for the triangle: ");
        int numberOfRows = scanner.nextInt();

        // Outer loop for the number of rows
        for (int i = 1; i <= numberOfRows; i++) {

            // Inner loop for printing spaces
            for (int j = 1; j <= numberOfRows - i; j++) {
                System.out.print("  ");
            }
```



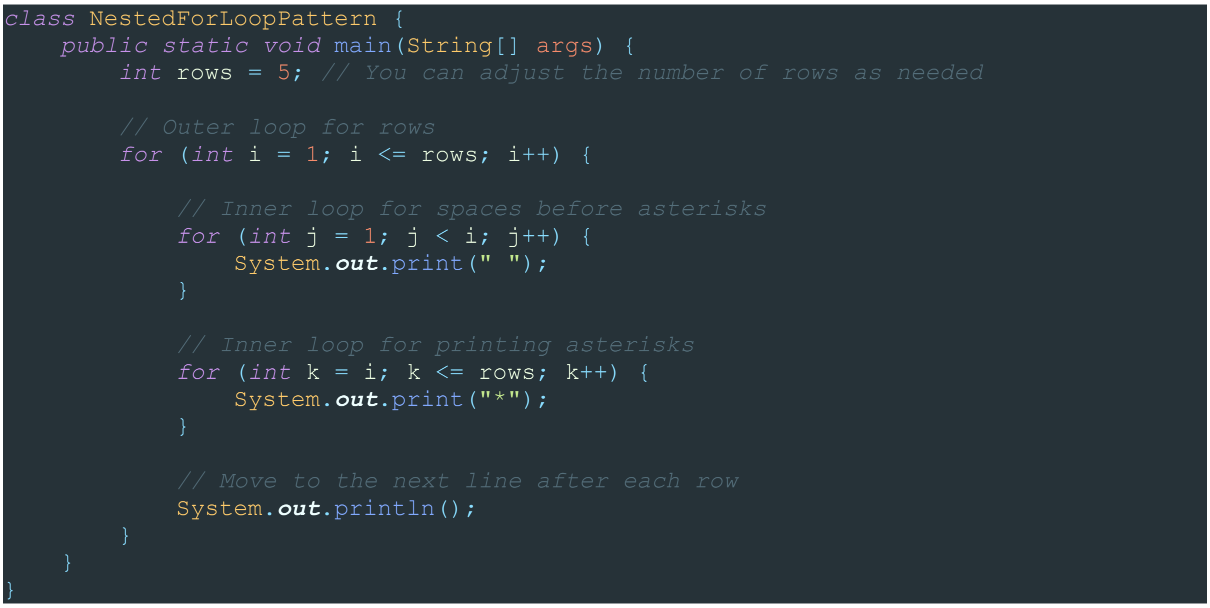
```
            // Inner loop for printing stars
            for (int k = 1; k <= i; k++) {
                System.out.print("* ");
            }

            // Move to the next line after printing stars in the row
            System.out.println();
        }

        scanner.close();
    }
}
```

Output:



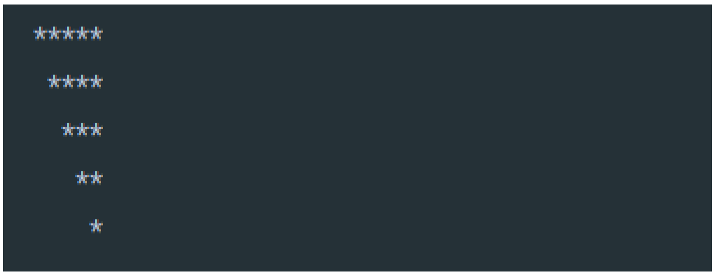
```
====Left Triangle Star Pattern===
Enter the number of rows for the triangle: 5
        *
      * *
    * * *
  * * * *
* * * * *
```

4.Inverted Left Triangle Star Pattern



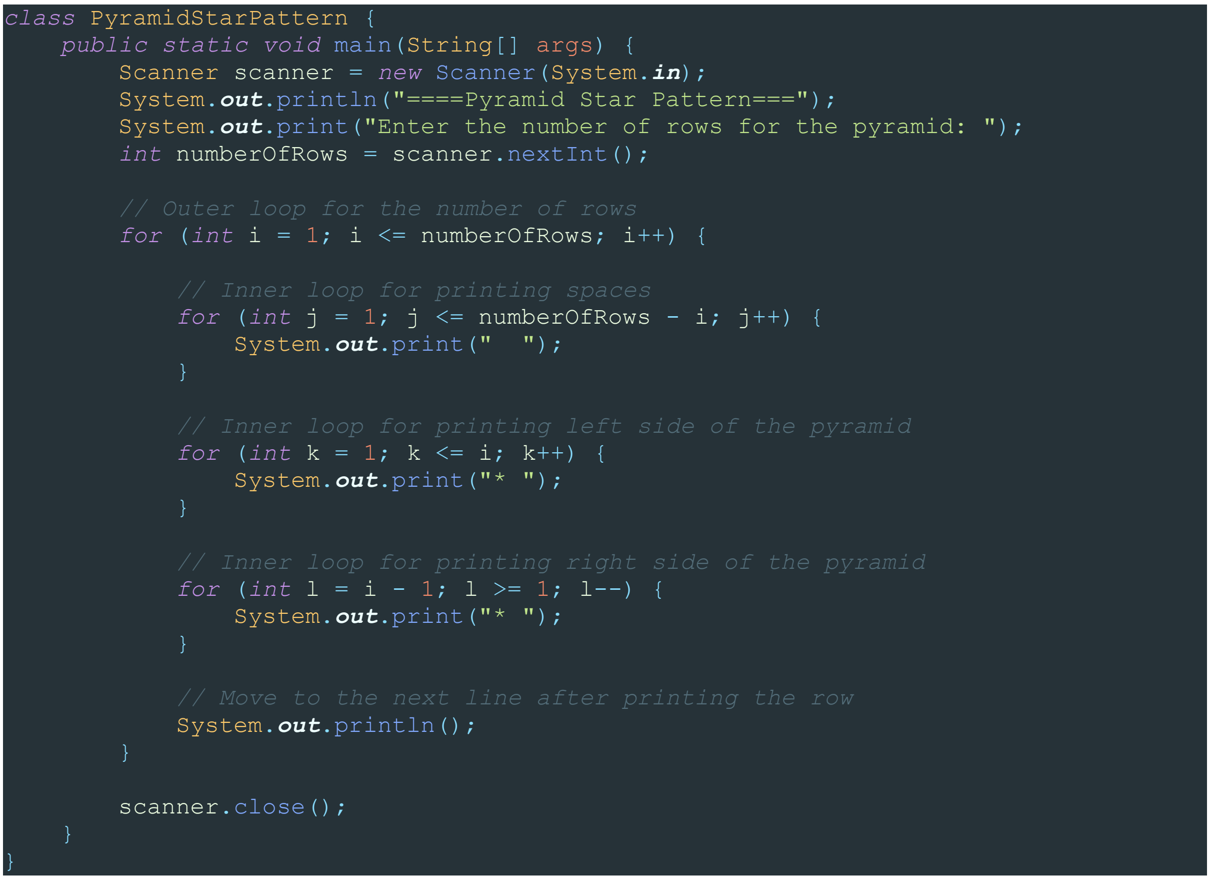
```
class NestedForLoopPattern {
    public static void main(String[] args) {
        int rows = 5; // You can adjust the number of rows as needed

        // Outer loop for rows
        for (int i = 1; i <= rows; i++) {

            // Inner loop for spaces before asterisks
            for (int j = 1; j < i; j++) {
                System.out.print(" ");
            }

            // Inner loop for printing asterisks
            for (int k = i; k <= rows; k++) {
                System.out.print("*");
            }

            // Move to the next line after each row
            System.out.println();
        }
    }
}
```

Output:



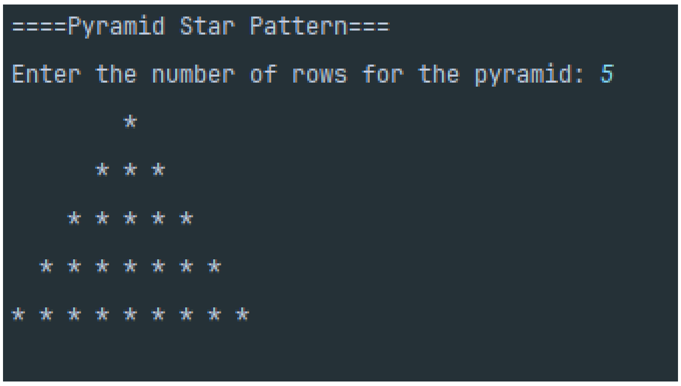
```
*****
 ****
  ***
   **
    *
```

5.Pyramid Star Pattern:



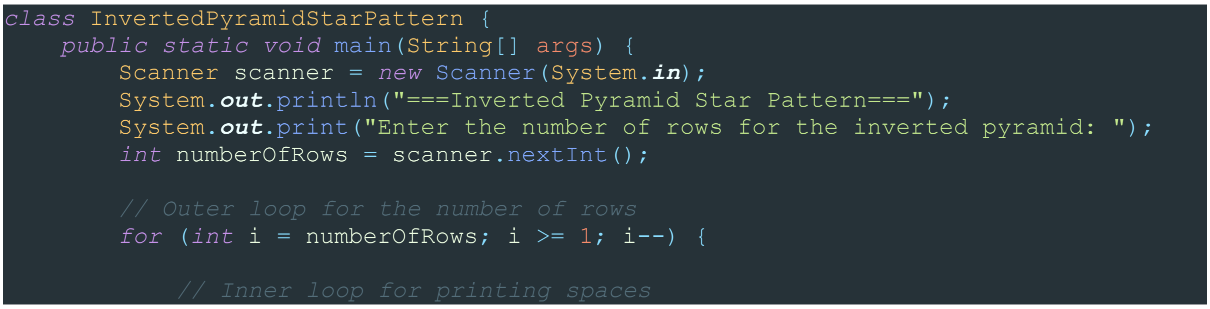
```
class PyramidStarPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("====Pyramid Star Pattern===");
        System.out.print("Enter the number of rows for the pyramid: ");
        int numberOfRows = scanner.nextInt();

        // Outer loop for the number of rows
        for (int i = 1; i <= numberOfRows; i++) {

            // Inner loop for printing spaces
            for (int j = 1; j <= numberOfRows - i; j++) {
                System.out.print("  ");
            }

            // Inner loop for printing left side of the pyramid
            for (int k = 1; k <= i; k++) {
                System.out.print("* ");
            }

            // Inner loop for printing right side of the pyramid
            for (int l = i - 1; l >= 1; l--) {
                System.out.print("* ");
            }

            // Move to the next line after printing the row
            System.out.println();
        }

        scanner.close();
    }
}
```

Output:

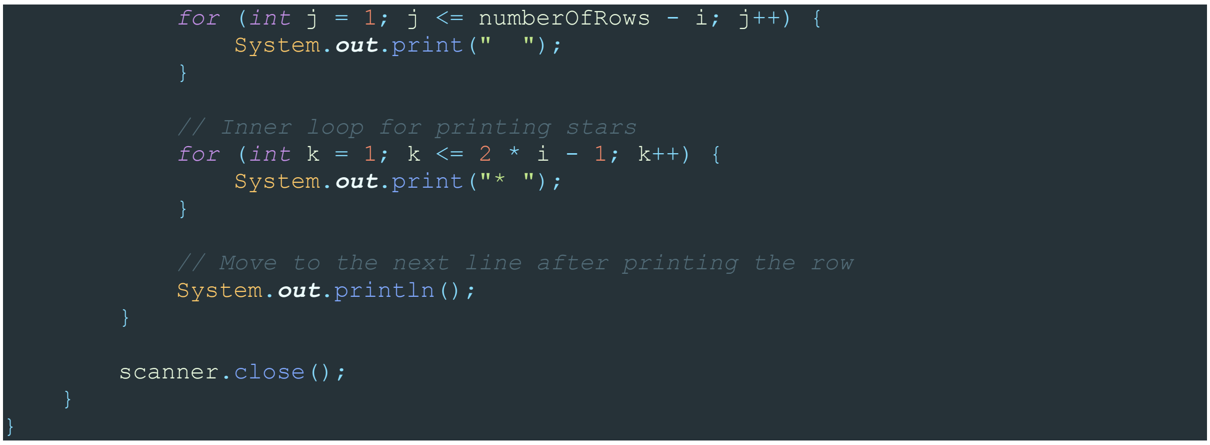


```
====Pyramid Star Pattern===
Enter the number of rows for the pyramid: 5
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

6.Inverted Pyramid Star Pattern:

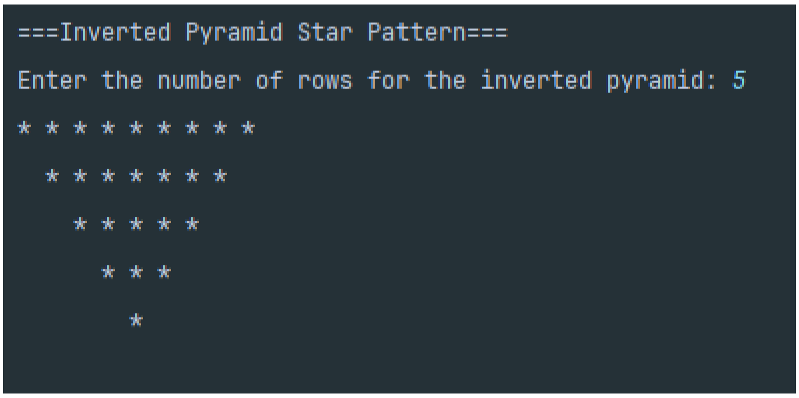


```
class InvertedPyramidStarPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("===Inverted Pyramid Star Pattern===");
        System.out.print("Enter the number of rows for the inverted pyramid: ");
        int numberOfRows = scanner.nextInt();

        // Outer loop for the number of rows
        for (int i = numberOfRows; i >= 1; i--) {

            // Inner loop for printing spaces
```



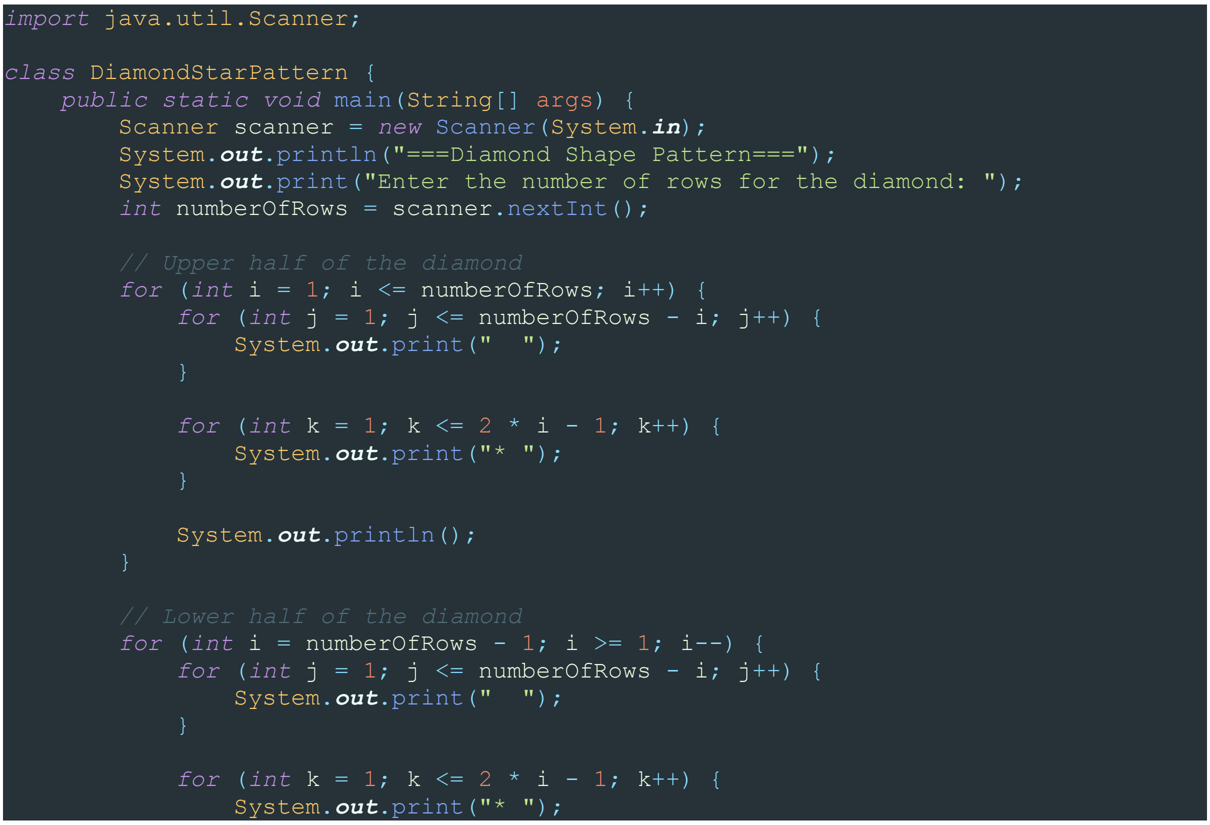
```
            for (int j = 1; j <= numberOfRows - i; j++) {
                System.out.print("  ");
            }

            // Inner loop for printing stars
            for (int k = 1; k <= 2 * i - 1; k++) {
                System.out.print("* ");
            }

            // Move to the next line after printing the row
            System.out.println();
        }

        scanner.close();
    }
}
```

Output:



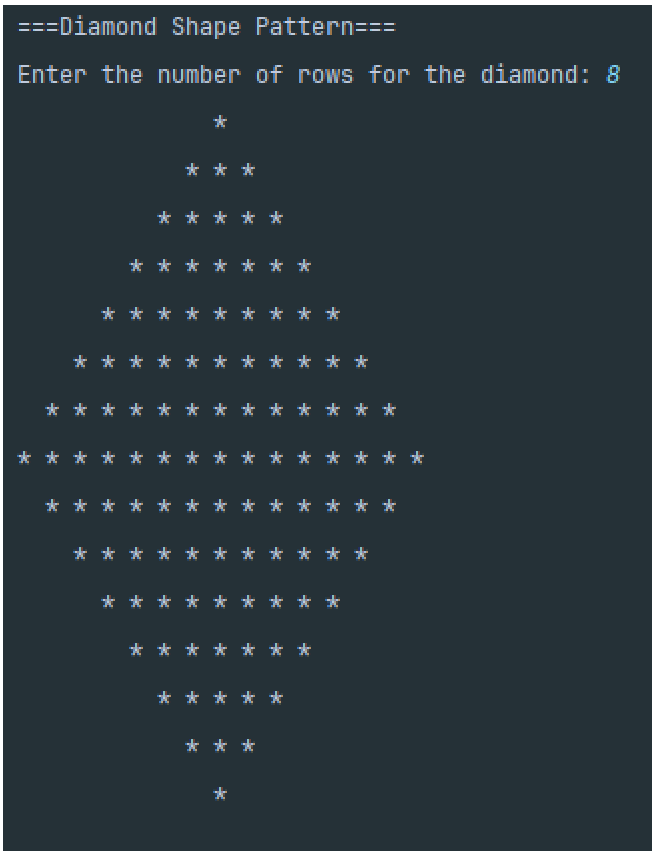
```
===Inverted Pyramid Star Pattern===

Enter the number of rows for the inverted pyramid: 5

* * * * * * * * *

  * * * * * * *

    * * * * *

      * * *

        *
```

7.Diamond Shape Pattern:



```
import java.util.Scanner;

class DiamondStarPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("===Diamond Shape Pattern===");
        System.out.print("Enter the number of rows for the diamond: ");
        int numberOfRows = scanner.nextInt();

        // Upper half of the diamond
        for (int i = 1; i <= numberOfRows; i++) {
            for (int j = 1; j <= numberOfRows - i; j++) {
                System.out.print("  ");
            }

            for (int k = 1; k <= 2 * i - 1; k++) {
                System.out.print("* ");
            }

            System.out.println();
        }

        // Lower half of the diamond
        for (int i = numberOfRows - 1; i >= 1; i--) {
            for (int j = 1; j <= numberOfRows - i; j++) {
                System.out.print("  ");
            }

            for (int k = 1; k <= 2 * i - 1; k++) {
                System.out.print("* ");
            }

            System.out.println();
        }

        scanner.close();
    }
}
```

Output:

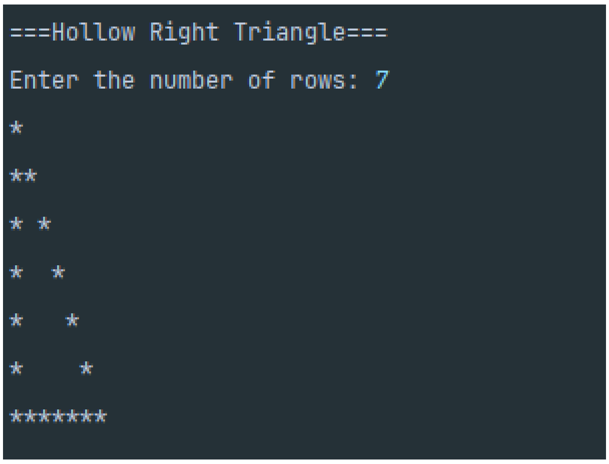


```
===Diamond Shape Pattern===
Enter the number of rows for the diamond: 8
              * 
            * * * 
          * * * * * 
        * * * * * * * 
      * * * * * * * * * 
    * * * * * * * * * * * 
  * * * * * * * * * * * * * 
* * * * * * * * * * * * * * * 
  * * * * * * * * * * * * * 
    * * * * * * * * * * * 
      * * * * * * * * * 
        * * * * * * * 
          * * * * * 
            * * * 
              * 
```

8.Hollow Inverted **Right Triangle Star Pattern**

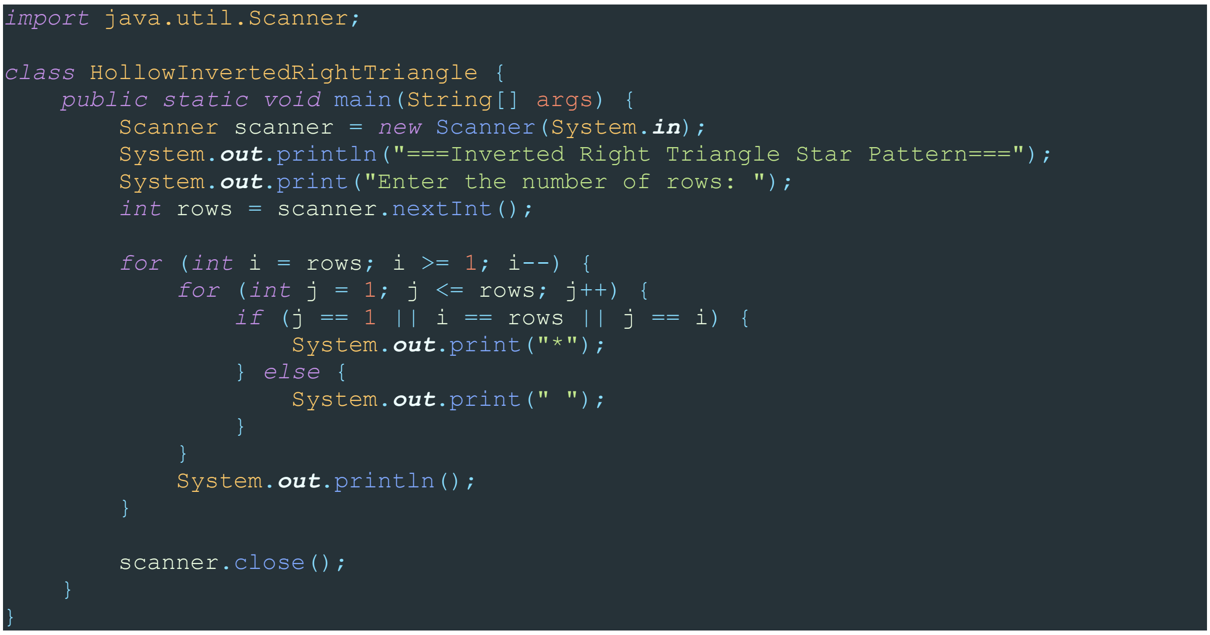
```
class HollowRightTriangle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("===Hollow Right Triangle===");
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                if (i == rows || j == 1 || j == i) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

Output:



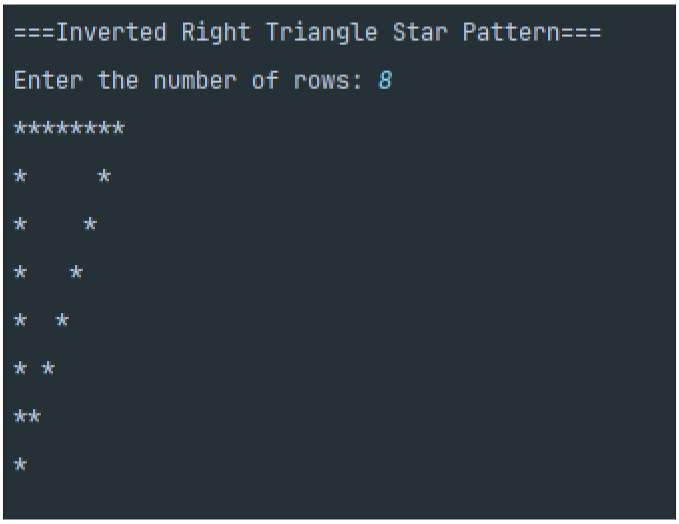
```
===Hollow Right Triangle===
Enter the number of rows: 7
*
**
* *
*  *
*   *
*    *
*******
```

9.Hollow Inverted Right Triangle:



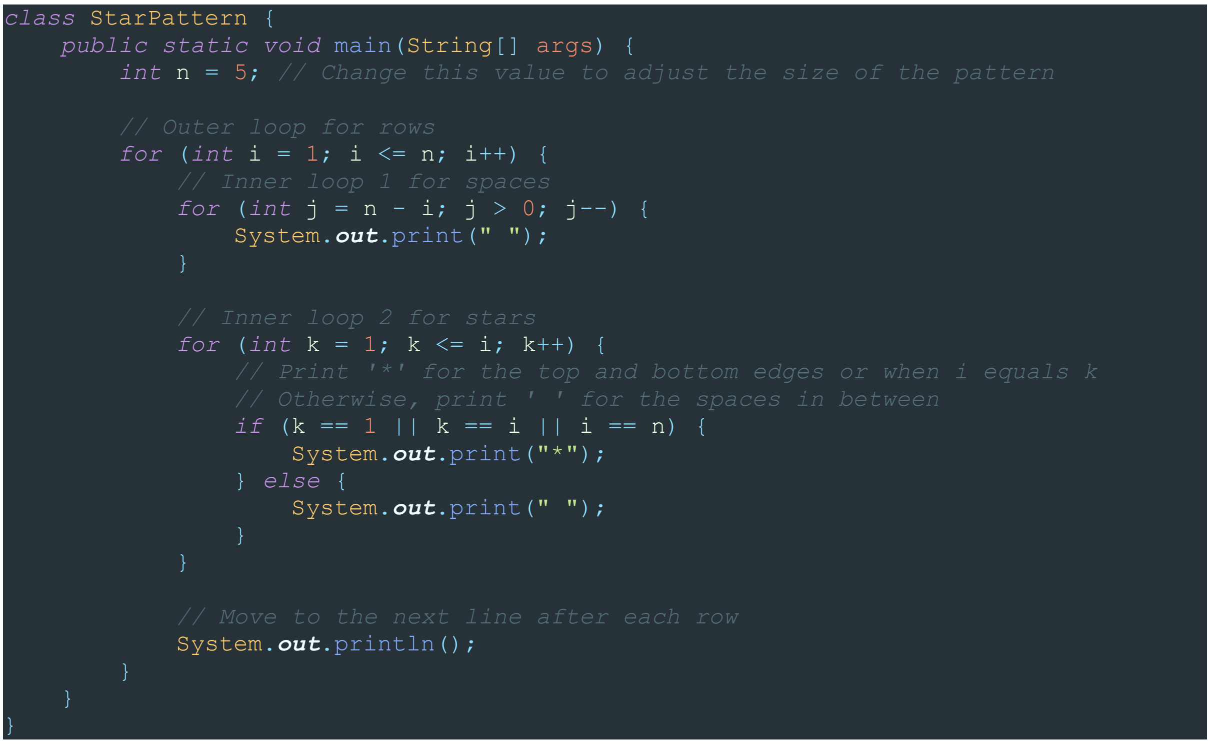
```
import java.util.Scanner;

class HollowInvertedRightTriangle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("===Inverted Right Triangle Star Pattern===");
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();

        for (int i = rows; i >= 1; i--) {
            for (int j = 1; j <= rows; j++) {
                if (j == 1 || i == rows || j == i) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

Output:



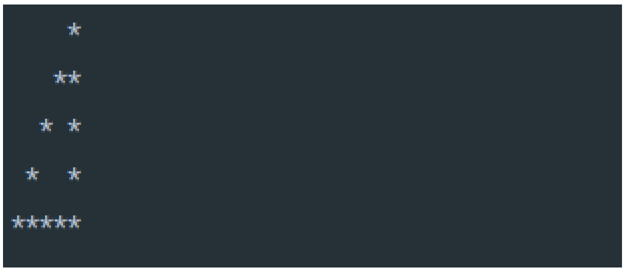
```
===Inverted Right Triangle Star Pattern===
Enter the number of rows: 8
********
*     *
*    *
*   *
*  *
* *
**
*
```

10.Hollow Left Triangle Star Pattern



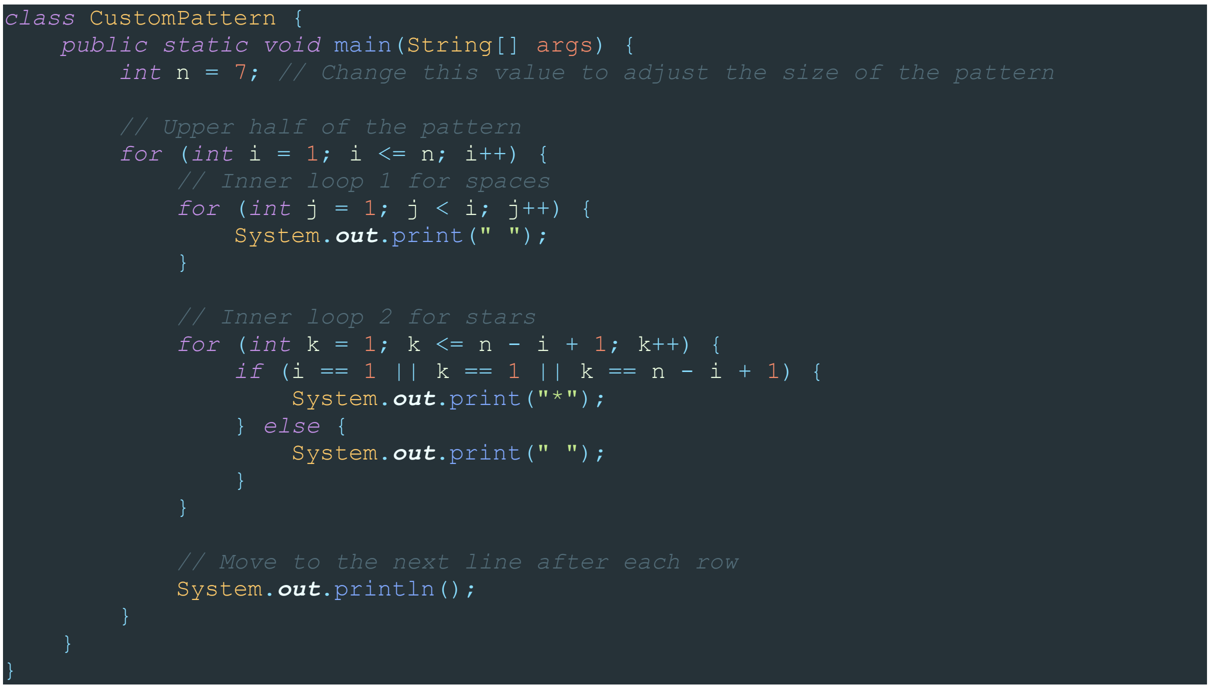
```java
class StarPattern {
    public static void main(String[] args) {
        int n = 5; // Change this value to adjust the size of the pattern

        // Outer loop for rows
        for (int i = 1; i <= n; i++) {
            // Inner loop 1 for spaces
            for (int j = n - i; j > 0; j--) {
                System.out.print(" ");
            }

            // Inner loop 2 for stars
            for (int k = 1; k <= i; k++) {
                // Print '*' for the top and bottom edges or when i equals k
                // Otherwise, print ' ' for the spaces in between
                if (k == 1 || k == i || i == n) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }

            // Move to the next line after each row
            System.out.println();
        }
    }
}
```

Output:



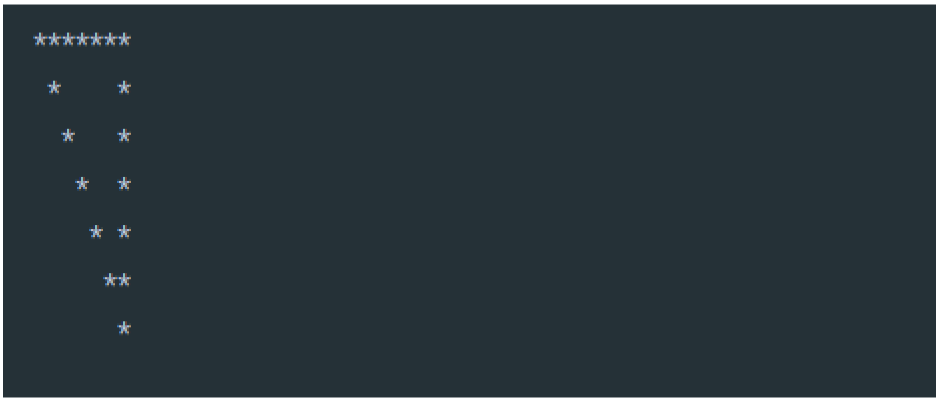
```
    *
   **
  * *
 *  *
*****
```

11.Hollow **Inverted Left Triangle Star Pattern**



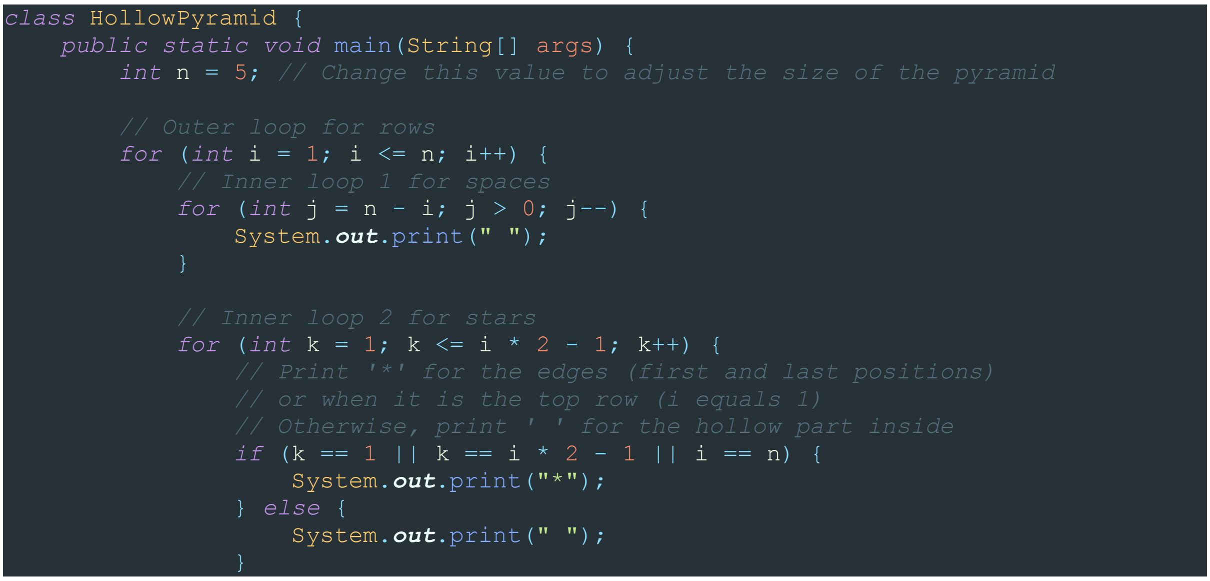
```java
class CustomPattern {
    public static void main(String[] args) {
        int n = 7; // Change this value to adjust the size of the pattern

        // Upper half of the pattern
        for (int i = 1; i <= n; i++) {
            // Inner loop 1 for spaces
            for (int j = 1; j < i; j++) {
                System.out.print(" ");
            }

            // Inner loop 2 for stars
            for (int k = 1; k <= n - i + 1; k++) {
                if (i == 1 || k == 1 || k == n - i + 1) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }

            // Move to the next line after each row
            System.out.println();
        }
    }
}
```

Output:

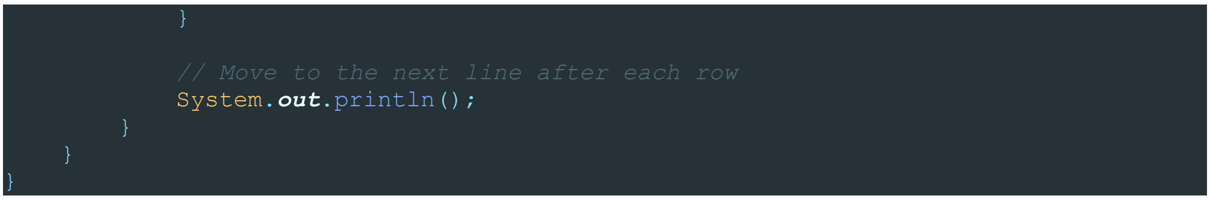


```
*******
 *    *
  *   *
   *  *
    * *
     **
      *
```

12. Hollow **Pyramid Star Pattern**

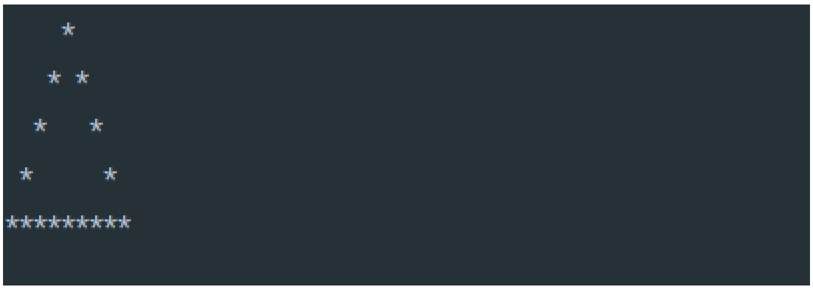


```java
class HollowPyramid {
    public static void main(String[] args) {
        int n = 5; // Change this value to adjust the size of the pyramid

        // Outer loop for rows
        for (int i = 1; i <= n; i++) {
            // Inner loop 1 for spaces
            for (int j = n - i; j > 0; j--) {
                System.out.print(" ");
            }

            // Inner loop 2 for stars
            for (int k = 1; k <= i * 2 - 1; k++) {
                // Print '*' for the edges (first and last positions)
                // or when it is the top row (i equals 1)
                // Otherwise, print ' ' for the hollow part inside
                if (k == 1 || k == i * 2 - 1 || i == n) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
```



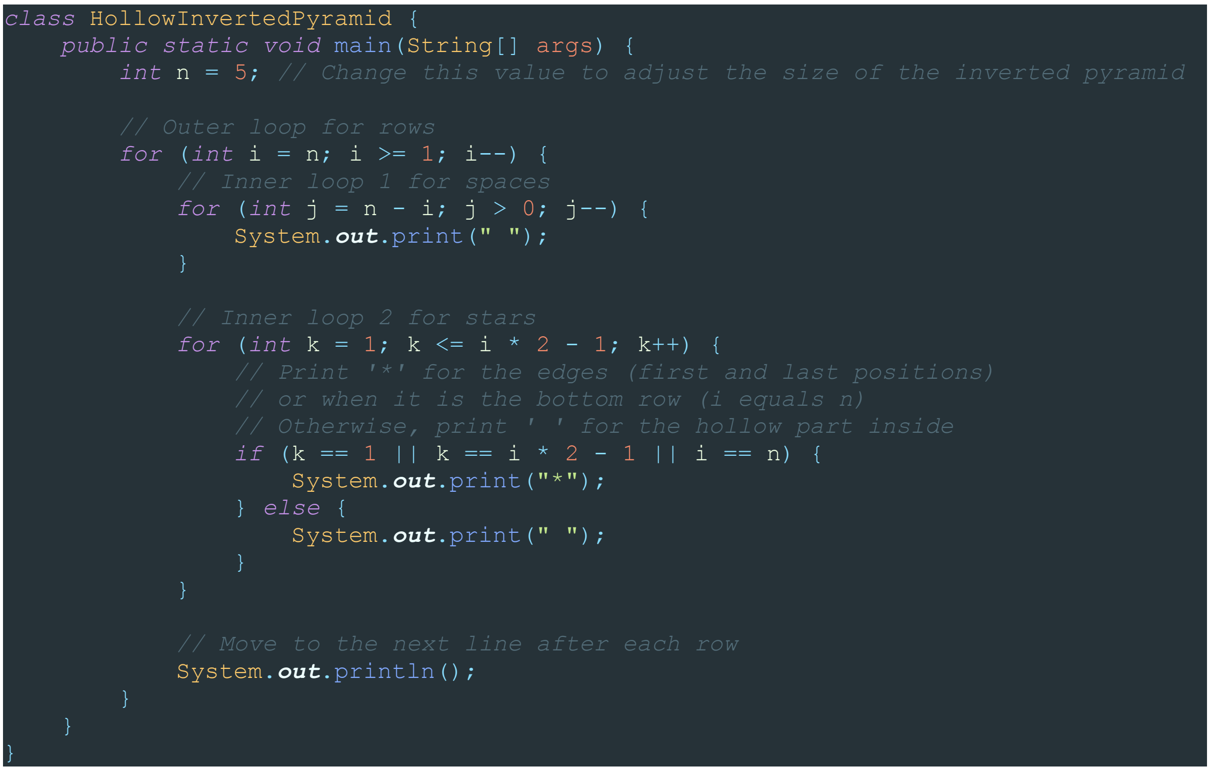
```
            }

            // Move to the next line after each row
            System.out.println();
        }
    }
}
```

Output:



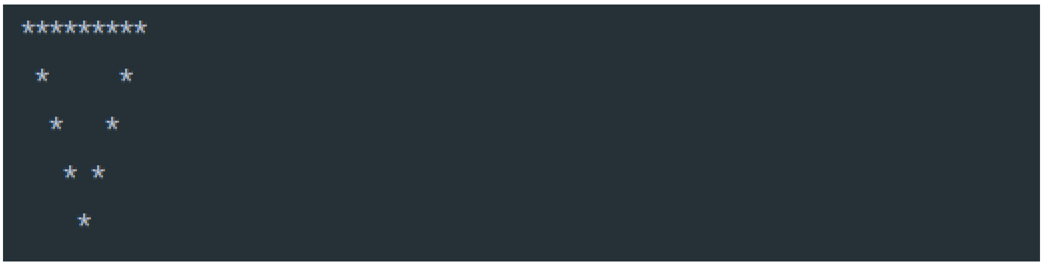
```
    *
   * *
  *   *
 *     *
*********
```

13.Hollow Inverted Pyramid Star Pattern



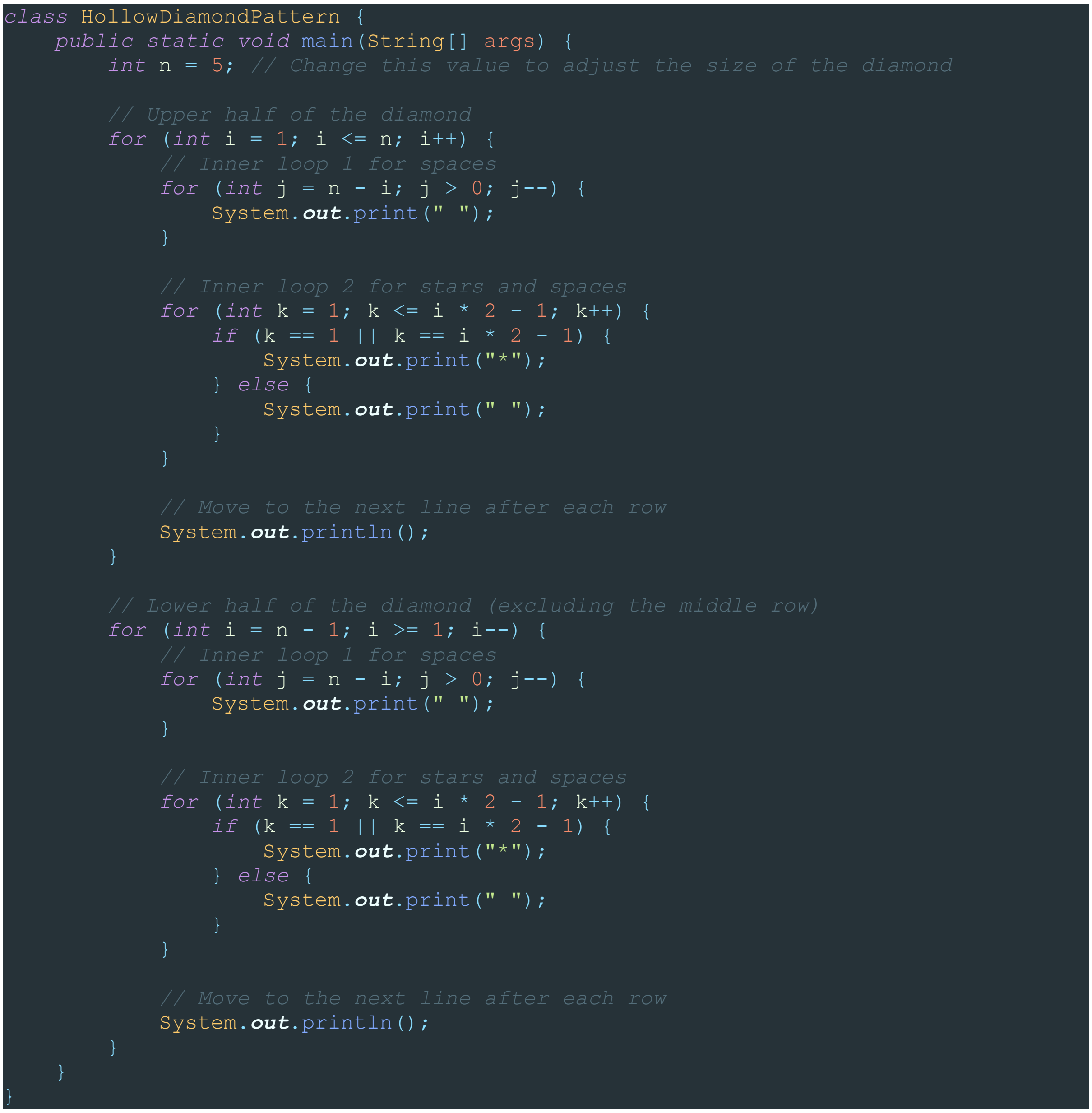
```
class HollowInvertedPyramid {
    public static void main(String[] args) {
        int n = 5; // Change this value to adjust the size of the inverted pyramid

        // Outer loop for rows
        for (int i = n; i >= 1; i--) {
            // Inner loop 1 for spaces
            for (int j = n - i; j > 0; j--) {
                System.out.print(" ");
            }

            // Inner loop 2 for stars
            for (int k = 1; k <= i * 2 - 1; k++) {
                // Print '*' for the edges (first and last positions)
                // or when it is the bottom row (i equals n)
                // Otherwise, print ' ' for the hollow part inside
                if (k == 1 || k == i * 2 - 1 || i == n) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }

            // Move to the next line after each row
            System.out.println();
        }
    }
}
```

Output:



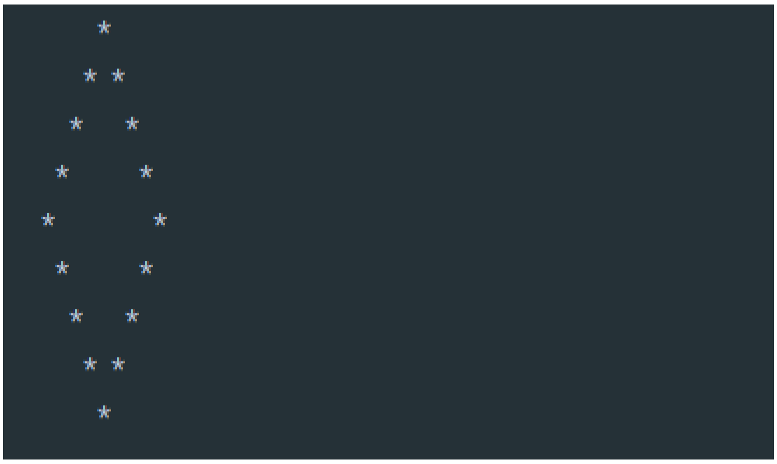
```
*********
 *     *
  *   *
   * *
    *
```

14.Hollow Diamond Star Pattern



```java
class HollowDiamondPattern {
    public static void main(String[] args) {
        int n = 5; // Change this value to adjust the size of the diamond

        // Upper half of the diamond
        for (int i = 1; i <= n; i++) {
            // Inner loop 1 for spaces
            for (int j = n - i; j > 0; j--) {
                System.out.print(" ");
            }

            // Inner loop 2 for stars and spaces
            for (int k = 1; k <= i * 2 - 1; k++) {
                if (k == 1 || k == i * 2 - 1) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }

            // Move to the next line after each row
            System.out.println();
        }

        // Lower half of the diamond (excluding the middle row)
        for (int i = n - 1; i >= 1; i--) {
            // Inner loop 1 for spaces
            for (int j = n - i; j > 0; j--) {
                System.out.print(" ");
            }

            // Inner loop 2 for stars and spaces
            for (int k = 1; k <= i * 2 - 1; k++) {
                if (k == 1 || k == i * 2 - 1) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }

            // Move to the next line after each row
            System.out.println();
        }
    }
}
```

Output:



```
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```